

Greedy is good:

An experimental study on minimum clique cover and maximum independent set problems for randomly generated rectangles

Ritankar Mandal* Anirban Ghosh[†] Sasanka Roy[‡] Subhas C. Nandy*

Abstract

Given a set $\mathcal{R} = \{R_1, R_2, \dots, R_n\}$ of n randomly positioned axis parallel rectangles in 2D, the problem of computing the minimum clique cover (MCC) and maximum independent set (MIS) for the intersection graph $G(\mathcal{R})$ of the members in \mathcal{R} are both computationally hard [4]. For the MCC problem, it is proved that polynomial time constant factor approximation is impossible to obtain [15]. Though such a result is not proved yet for the MIS problem, no polynomial time constant factor approximation algorithm exists in the literature. We study the performance of greedy algorithms for computing these two parameters of $G(\mathcal{R})$. Experimental results shows that for each of the MCC and MIS problems, the corresponding greedy algorithm produces a solution that is very close to its optimum solution. Scheinerman [16] showed that the size of MIS is tightly bounded by \sqrt{n} for a random instance of the 1D version of the problem, (i.e., for the interval graph). Our experiment shows that the size of independent set and the clique cover produced by the greedy algorithm is at least $2\sqrt{n}$ and at most $3\sqrt{n}$, respectively. Thus the experimentally obtained approximation ratio of the greedy algorithm for MIS problem is at most $\frac{3}{2}$ and the same for the MCC problem is at least $\frac{2}{3}$. Finally we will provide refined greedy algorithms based on a concept of *simplicial rectangle*. The characteristics of this algorithm may be of interest in getting a provably constant factor approximation algorithm for random instance of both the problems. We believe that the result also holds true for any finite dimension.

Keywords: Minimum clique cover, maximum independent set, rectangle intersection graph, approximation algorithm, empirical study

1 Introduction

Let $G(\mathcal{R})$ be the intersection graph of a set $\mathcal{R} = \{R_1, R_2, \dots, R_n\}$ of n randomly placed polygonal objects, e.g., rectangles, circles, polygons, etc., in a 2D region. In the practical applications two types of cliques in the geometric intersection graph are considered, namely (i) graphical clique and (ii) geometric clique. A graphical clique C in $G(\mathcal{R})$ is a maximal complete subgraph of $G(\mathcal{R})$. A geometric clique C' consists of a maximal set of objects $C' \subseteq \mathcal{R}$ such that they have a common point in their interior. Thus, a geometric clique is always a graphical clique, however the converse is not true. If \mathcal{R} is a set of polygonal objects, then the problem of computing the minimum geometric clique cover is NP-hard [6]. However, for a set \mathcal{R} of axis-parallel rectangles, a graphical clique is always a geometric clique since the axis-parallel rectangles satisfy Helly property. Thus,

*Indian Statistical Institute, Kolkata 700108, India

[†]Dept. of CS, University of Wisconsin-Milwaukee, Milwaukee, WI 53201, USA

[‡]Chennai Mathematical Institute, Chennai, India

the minimum clique cover of the graph $G(\mathcal{R})$ is same as the minimum number of points required to stab all the rectangles in $\hat{\mathcal{R}}$. It is easy to show that the number of cliques present in $G(\mathcal{R})$ may be $O(n^2)$ in the worst case [11]. But, if \mathcal{R} is a set of c -oriented polygons ($c \geq 5$), the Helly property does not hold. Nilson [14] proved that the number of geometric clique in $G(\mathcal{R})$ can be at most $\tau(2, c)\phi(\mathcal{R})\log_2^{c-1}(\phi(\mathcal{R}) + 1)$, where $\tau(2, c)$ is the Gallai number of the pairwise intersecting c -oriented polygons and Let $\phi(\mathcal{R})$ denotes the packing number of \mathcal{R} , that is the maximum number of pairwise disjoint objects in \mathcal{R} . The same paper also provides a $t(n, c) + O(nc\log(\phi(\mathcal{R})))$ time algorithm for computing the minimum geometric clique cover of $G(\mathcal{R})$, where $t(n, c)$ is the time required to pierce pairwise intersecting c -oriented polygons.

In this note, we are interested in studying the performance of the greedy algorithm for computing the minimum clique cover (MCC) and maximum independent set (MIS) of a set $G(\mathcal{R})$ of axis-parallel rectangles in 2D. The study on MCC and MIS problems on a random instance of interval graph have started long back by Scheinerman [16]. They showed a tight bound of \sqrt{n} for both MCC and MIS problems on randomly generated intervals. Recently, Tran [17] formally proved that size of the MCC for $G(\mathcal{R})$ can be at most $O(\sqrt{n}\log\log n)$. He conjectured that size of the MCC is at most $O(\sqrt{n})$ for $G(\mathcal{R})$. He also provided results for higher dimension of the problem. A lot of studies on MCC and MIS problems have been done for the intersection graph of a set of axis-parallel rectangles on a 2D plane. Aronov et al. [2] proposed a randomized polynomial time algorithm for the MCC problem using the concept of ϵ -net that can produce $O(\log\log n)$ factor approximation result. Later Pach et al. [15] showed that this is the best possible approximation for the MCC problem that can be achieved in polynomial time. For the MIS problem, the first approximation algorithm for arbitrary sized axis-parallel rectangles was proposed by Agarwal et al. [1] that produces a $O(\log n)$ -factor approximation result in $O(n\log n)$ time. The best known result for MIS problem is due to Chalermsook and Chuzhoy [3], which provides a polynomial time $O(\log\log n)$ factor approximation algorithm for the said problem. A nice literature on MIS problem can be found in [3]. Although there is a lower-bound proof on approximation ratio of MCC problem, same for MIS problem is not known. Nielson [13] gives a construction which gives $\Omega(\log n)$ bound for the greedy algorithm posed in [5] (see Figure 1).

Halldórsson and Radhakrishnan [8] showed that for general graphs of bounded degree Δ , the greedy algorithm produces $\frac{\Delta+1}{3}$ -factor approximation result for the MIS problem. It also proposes a simple parallel algorithm that runs in $O(\log^* n)$ time using linear number of processors. The approximation factor can be improved to $\frac{2\bar{d}+3}{5}$ using the fractional relaxation technique of Nemhauser and Trotter [12], where \bar{d} is the average degree of the graph. Finally, it shows that using the greedy strategy of removing all cliques of same size gradually improves the approximation ratio of the algorithm, and $\frac{\Delta}{3.76}$ approximation factor is possible to achieve. Mestre [10] introduced the notion of k -extendible systems and showed that for such a system the greedy algorithm produces a $\frac{1}{k}$ -factor approximation result. They showed that maximum weight b -matching forms a 2-extendible system, and hence greedy algorithm produces a $\frac{1}{2}$ factor approximation result. Several other problems, namely maximum profit scheduling, maximum asymmetric TSP, can be shown to satisfy the properties of k -extendible system for some suitable k .

2 Our contribution

The main contribution of this paper is an empirical study showing that the greedy algorithms for each of the MCC and MIS problems produces very close result when compared with optimum solution for a random instance of the intersection graph of axis-parallel rectangles. Our experiment pushes the conjecture made by Tran [17] more towards affirmative side because the size of the clique cover produced by the greedy algorithm for the MCC problem on a random instance of rectangle

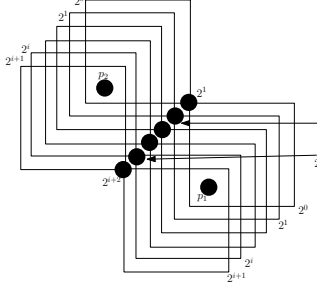


Figure 1: An example showing that approximation factor of the greedy algorithm is $\log n$

intersection graph is at most $3\sqrt{n}$ for reasonably big random instances and we believe that it will hold true for any such instances. Similarly, the greedy algorithm for the MIS problem also produces a very close solution to the optimum solution in the sense that the size of MIS produced by greedy algorithm is at least $2 \times \sqrt{n}$. Finally we will provide two refined greedy algorithms and their characteristics that may be of interest to design constant factor approximation algorithms for the MCC and MIS problems where the participating rectangles are randomly placed and of random size. We strongly believe that the result also holds for the intersection graph of randomly generated axis-parallel rectangles in any finite dimension. We have also produced the result of Nielson's [13] divide and conquer algorithm for the MCC problem. Nielson [13] has done some experimental study of his algorithm on the intersection graph of randomly generated rectangles and claimed that the approximation ratio of their algorithm is 3.42. We show that the size of the clique cover produce by our greedy algorithm is almost 2 times better than that of [13].

3 Algorithms

We generate a set $\mathcal{R} = \{R_1, R_2, \dots, R_n\}$ of axis-parallel rectangles in a given rectangular region $([a, b], [c, d])$. Each rectangle is stored in the form of a pair of points indicating its bottom-left and top-right corners. In other words, for the rectangle R_i , we randomly choose a pair of points $p_i = (x_p, y_p)$, $q_i = (x_q, y_q)$ on the given rectangular region $([a, b], [c, d])$. Without loss of generality assume that $x_p < x_q$. Now, if $y_p < y_q$, then $R_i = [(x_p, y_p), (x_q, y_q)]$; otherwise $R_i = [(x_p, y_q), (x_q, y_p)]$. Now, we introduce the concept of *dominated rectangle*. If a rectangle R_i contains some other rectangle R_j , $j \neq i$, then it is called a dominated rectangle. A dominated rectangle can be disregarded since it need not be considered while computing the minimum clique cover, and also it does not participate in the maximum independent set. After the generation of n random rectangles, let $\hat{\mathcal{R}}$ denote the set of all non-dominated rectangles.

We use $G(\hat{\mathcal{R}})$ to denote the intersection graph of the members in $\hat{\mathcal{R}}$. Here the nodes correspond to the members in $\hat{\mathcal{R}}$. Between a pair of vertices there is an edge if the corresponding two rectangles share a common point in their interior. We use $N(R_i)$ to denote the vertices adjacent to R_i in $G(\hat{\mathcal{R}})$ including itself. We now describe the two greedy heuristic algorithms for the MCC and MIS problems. We have also implemented the divide and conquer algorithm of Nielson [13] for the MCC problem to justify the approximation bound of our algorithms on the MCC problem for a random instant of rectangle intersection graph. We use this approximation bound to justify the approximation bound of our greedy algorithm for the MIS problem for a random instant of rectangle intersection graph.

Algorithm 1 Algorithm $GCC(\mathcal{R})$

- 1: **Input:** A set \mathcal{R} of randomly generated n rectangles on a 2-dimensional plane.
 - 2: **Output:** A set GCC of points that stab all the rectangles in \mathcal{R} .
 - 3: Compute $\hat{\mathcal{R}}$, the non-dominated set of rectangles in \mathcal{R} .
 - 4: Initialize $GCC = \emptyset$.
 - 5: **repeat**
 - 6: Compute a maximum clique \mathcal{C} among the rectangles in $\hat{\mathcal{R}}$ using the algorithm of [9, 11], and choose a point π in the common region of the members in \mathcal{C} .
 - 7: Set $\hat{\mathcal{R}} = \hat{\mathcal{R}} \setminus \mathcal{C}$ and $GCC = GCC \cup \{\pi\}$.
 - 8: **until** $\hat{\mathcal{R}} = \emptyset$.
 - 9: Return GCC as the stabbing points along with its cardinality.
-

3.1 MCC problem

Since the axis-parallel rectangles satisfy Helly property, the members of a clique in $G(\mathcal{R})$ share a point in their interior. Thus, the minimum clique cover of the graph $G(\mathcal{R})$ is same as the minimum number of points required to stab all the rectangles in $\hat{\mathcal{R}}$.

3.1.1 Greedy algorithm

Our greedy algorithm proceeds as follows. We perform a horizontal line sweep from top to bottom to compute the largest clique \mathcal{C} of $G(\mathcal{R})$. It is a point where maximum number of rectangles in $\hat{\mathcal{R}}$ overlap. All these rectangles can be stabbed by a single point. We delete all the rectangles in \mathcal{C} from $\hat{\mathcal{R}}$ and repeat the same process. The iteration continues until all the rectangles in $\hat{\mathcal{R}}$ are deleted. The pseudo code of the algorithm is given below.

Lemma 1. *The worst case time complexity of the algorithm GCC is $O(n^2 \log n)$.*

Proof. Follows from the fact that the largest clique of a rectangle intersection graph can be computed in $O(n \log n)$ time [9, 11], and the number of iterations can be $O(n)$ in the worst case. \square

3.1.2 An improved greedy algorithm

We now introduce the concept of *simplicial rectangle* to present an improvement of the greedy algorithm GCC_I for the MCC problem. A node in the graph $G(\mathcal{R})$ is said to be simplicial if all the nodes adjacent to it form a clique. The corresponding rectangle will be referred to as *simplicial rectangle*. The concept of simplicial rectangle is very much similar to a simplicial vertex in a graph [7]. It needs to be noted that all the rectangles adjacent to a simplicial rectangle can be stabbed by a point along with the rectangle R . Algorithm 2 states the detailed procedure.

Lemma 2. *The worst case time complexity of the algorithm GCC_I is $O(n^3)$.*

Proof. Each iteration of the repeat loop of the algorithm consists of two steps: (i) finding a simplicial rectangle, and (ii) finding the largest clique.

Let $N(R)$ denote the neighbors of R including itself in the graph $G(\mathcal{R})$. In Step (i), the construction of the incidence matrix I needs $O(n^2)$ time. While searching for a simplicial rectangle, in each failure step it marks all the members of $N(R)$ since there exists no other rectangle $R' \in N(R)$ which is simplicial. The proof is as follows. If $N(R) = N(R')$ then surely R' is not simplicial. Otherwise there exists some rectangle R'' such that $R'' \in N(R')$ but $R'' \notin N(R)$; since $R \in N(R')$, R' is

Algorithm 2 Algorithm GCC-I(\mathcal{R})

```
1: Input: A set  $\mathcal{R}$  of  $n$  randomly generated rectangles on a 2D plane.
2: Output: A set  $GCCI$  of points that stab all the rectangles in  $\mathcal{R}$ .
3: Compute the non-dominated set of rectangles  $\hat{\mathcal{R}}$ .
4: Initialize  $\Theta = \emptyset$ ;  $\Phi = \emptyset$ .
5: repeat
6:    $R = \text{Find\_Simplicial}(\hat{\mathcal{R}})$ .
7:   if  $R \neq 0$  then
8:     Let  $N(R)$  be the rectangles adjacent to  $R$  including itself.
9:     Choose a point  $\pi$  in the region common to all the members in  $N(R)$ .
10:    Set  $\hat{\mathcal{R}} = \hat{\mathcal{R}} \setminus N(R)$  and  $\Theta = \Theta \cup \{\pi\}$ .
11:   else
12:     Compute a maximum clique  $\mathcal{C}$  among the rectangles in  $\hat{\mathcal{R}}$  using the algorithm of [9, 11], and choose
        a point  $\pi$  in the common region of the members in  $\mathcal{C}$ .
13:     Set  $\hat{\mathcal{R}} = \hat{\mathcal{R}} \setminus \mathcal{C}$  and  $\Phi = \Phi \cup \{\pi\}$ .
14:   end if
15: until  $\hat{\mathcal{R}} = \emptyset$ 
16: return  $GCCI = \Theta \cup \Phi$  as the stabbing points along with its cardinality.
```

```
1: Procedure Find_Simplicial( $A$ )
2: Assumption: All the vertices in  $A$  are unmarked.
3: Compute an incidence matrix  $I$  of the graph  $G(A)$ ;
4: Sort the vertices of  $G(A)$  in increasing order of their degrees;
5: while all the vertices in  $A$  are not marked do
6:   choose the vertex  $v$  having minimum degree among the unmarked vertices.
   Let  $N(v)$  be the set of vertices adjacent to  $v$ .  $N(v)$  includes the vertex  $v$ .
7:   if  $N(v)$  forms a clique then
8:     return  $R$  as a simplicial rectangle, and a point  $\pi$  inside the common intersection region;
9:   else
10:    Mark the members of  $N(v)$  since they can not be simplicial.
11:    if  $a, b \in N(v)$  and  $I[a, b] = 0$  then
12:      mark all the vertices  $u$  such that  $I[a, u] = 1$  and  $I[b, u] = 1$ 
13:    end if
14:  end if
15: end while
16: return 0.
```

not simplicial. Moreover, if $\rho, \rho' \in N(R)$ and $I(\rho, \rho') = 0$, then we have deleted all the rectangles having both ρ and ρ' as neighbors. Thus, each entry of the matrix I is accessed $O(1)$ time.

Step (ii) needs $O(n \log n)$ time in the worst case [9, 11]. Since the number of iterations is $O(n)$ in the worst case, the result follows. \square

3.2 MIS problem

3.2.1 Greedy algorithm

Our MIS heuristic also depends on the concept of simplicial rectangle. If a simplicial rectangle R is found in $G(\hat{\mathcal{R}})$ (i.e., $N(R)$ forms a clique), we can only choose R in the independent set among the set of rectangles $N(R)$. Our algorithm is an iterative one. At each iteration, it searches for a simplicial rectangle. If such a rectangle R is found, it is included in MIS; otherwise, we delete a rectangle having maximum number of neighbors. The logic behind choosing such a rectangle is

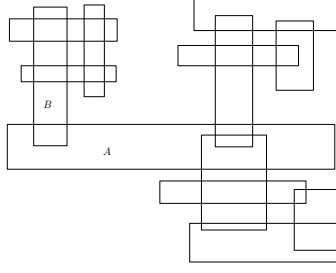


Figure 2: The maximum degree rectangle removal may lead to non-optimal result

that its absence may delete a neighbor of maximum number of rectangles. Note that, this may also be a simplicial rectangle if some of its adjacent rectangle is removed. In Figure 2 such a situation is demonstrated. Here none of the rectangles present in the region is simplicial due to the position of the other rectangles. Rectangle A has maximum number of neighbors. But removal of rectangle B makes it simplicial. However, the chance of such a rectangle to be simplicial is small. The pseudo code of the algorithm is given in Algorithm 3.

Algorithm 3 Algorithm MIS(\mathcal{R})

- 1: **Input:** A set \mathcal{R} of n randomly generated rectangles on a 2D plane.
 - 2: **Output:** A set MIS of mutually non-overlapping rectangles.
 - 3: Compute $\hat{\mathcal{R}}$, the non-dominated set of rectangles in \mathcal{R} .
 - 4: Initialize $MIS = \emptyset$.
 - 5: **repeat**
 - 6: $R = \text{Find_Simplicial}(\hat{\mathcal{R}})$.
 - 7: **if** $R \neq 0$ **then**
 - 8: Let $N(R)$ be the set of rectangles adjacent to R including itself.
 - 9: Set $\hat{\mathcal{R}} = \hat{\mathcal{R}} \setminus N(R)$ and $MIS = MIS \cup \{R\}$.
 - 10: **else**
 - 11: Let R' be the rectangle having maximum degree.
 - 12: Set $\hat{\mathcal{R}} = \hat{\mathcal{R}} \setminus \{R'\}$
 - 13: **end if**
 - 14: **until** $\hat{\mathcal{R}} = \emptyset$
 - 15: Return MIS along with its cardinality.
-

Lemma 3. *The worst case time complexity of the algorithm MIS is $O(n^3)$.*

Proof. Each iteration of the repeat loop of the algorithm consists of two steps: (i) finding a simplicial rectangle, and (ii) finding the rectangle having maximum degree. The result follows from the time complexity of Step (i) (see Lemma 2). \square

3.2.2 A variation of the Greedy algorithm

In our greedy algorithm (subsection 3.2.1), at each iteration we searched for a simplicial rectangle. If such a rectangle is not found, then we removed the rectangle having maximum degree, and repeated the process.

The algorithm proposed in this section is very similar to *GCCI* algorithm. Here instead of removing the rectangle having maximum degree, we identified the largest clique and then removed all the rectangles participating in that clique. Finally, we report the set simplicial rectangles as the independent set. The time complexity of this algorithm remains same as that of *GCCI*.

4 Experimental Studies

We have performed a detailed experiment with different n (number of rectangles). For each n , we generated 20 different instances of n random rectangles as described in Section 3. For each instance, we run our proposed heuristics for both MCC and MIS problem, and also the divide and conquer heuristics of [13] for the MCC problem. In Table 1 and Figure 3, we refer this algorithm as DCC. Figure 3 shows the comparison of performance of our proposed two greedy heuristics GCC and GCC_I and the divide and conquer algorithm DCC of [13] for the MCC problem on rectangle intersection graph. It is observed that our both the algorithms produce result better than that of [13]. We have also plotted $3\sqrt{n}$ for different values of n in the same graph to demonstrate the solution produced by the improved greedy heuristic GCC_I is always less than $3\sqrt{n}$.

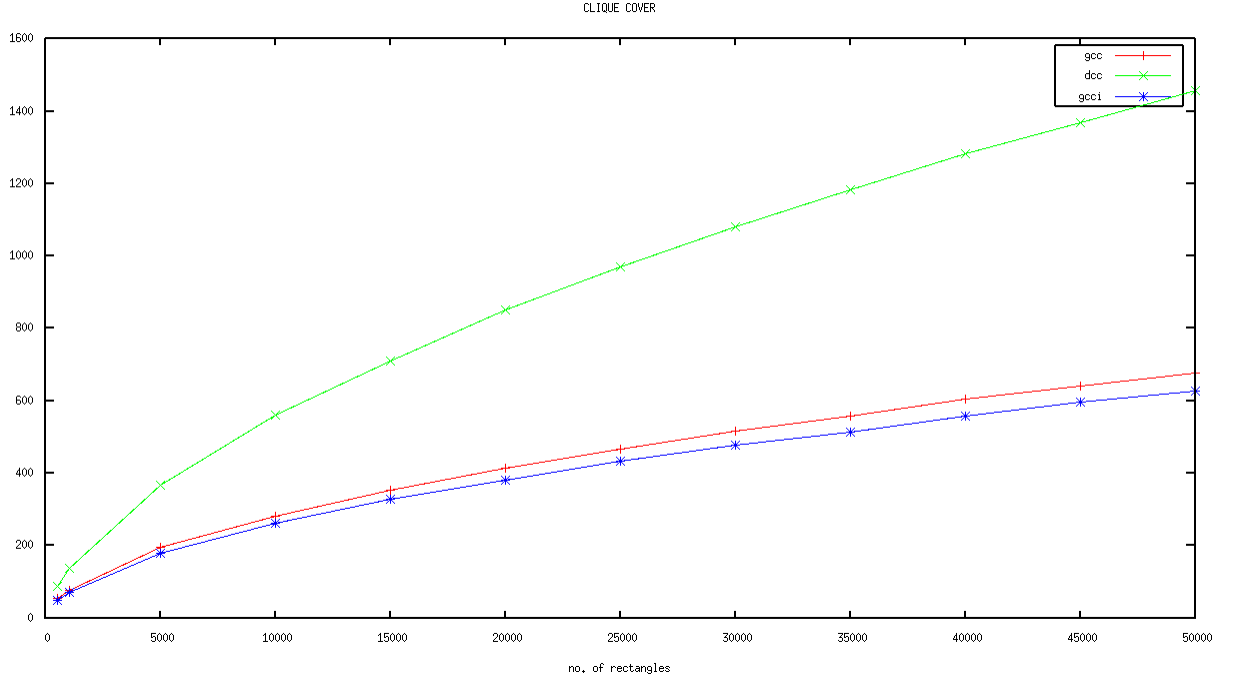


Figure 3: Comparison of the quality of solution produced by our proposed heuristics GCC , GCC_I and the DCC algorithm of [13] for the clique cover of rectangle intersection graph

In Figure 4, we demonstrate the performance of our proposed two greedy heuristics MIS and MIS_I for the maximum independent set problem on rectangle intersection graph. We have also plotted $2\sqrt{n}$ for different values of n in the same graph to demonstrate the solution produced by the greedy heuristic MIS is always less than $2\sqrt{n}$. But, the running time of our MIS_I heuristic is much better than that of MIS .

The final conclusion of our experimental study is summarized in Observation 1, and is demonstrated in Figure 5, The justifications of getting such results are also explained.

Observation 1. *The solution produced by our greedy heuristics for the minimum clique cover (MCC) problem on an intersection graph of a set of randomly generated axis-parallel rectangles is at most $2 \times OPT_{MCC}$, where OPT_{MCC} is the size of the optimum solution of the same problem.*

Justification 1: Let OPT_{MIS} be the size of the optimum solution of the maximum independent set (MIS) problem. We have $\frac{OPT_{MIS}}{OPT_{MCC}} \leq 1$. Figure 5 shows that $\frac{|GCC_I|}{|MIS|} \leq 1.5 < 2$ for all

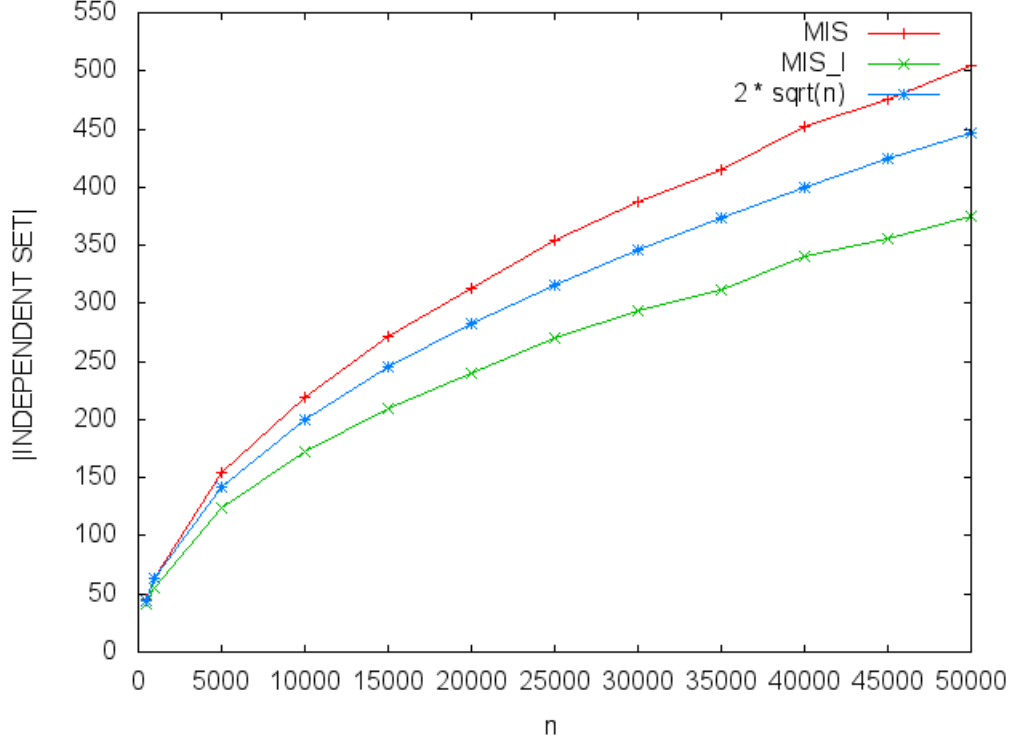


Figure 4: Comparison of the results produced by our proposed heuristics MIS and MIS_I for the maximum independent set of a rectangle intersection graph

Table 1: Experimental result

n	min. clique cover			max. indep. set		GCC_I/MIS
	DCC[13]	GCC	GCC_I	MIS	MIS_I	
500	87	53	48	46	41	1.0434
1000	137	77	71	64	55	1.1094
5000	367	195	180	155	124	1.1613
10000	560	281	262	219	172	1.1963
15000	709	353	327	271	210	1.2066
20000	851	414	382	313	240	1.2204
25000	971	467	432	354	270	1.2203
30000	1080	515	478	388	294	1.2320
35000	1183	557	514	415	311	1.2386
40000	1282	604	559	452	341	1.2367
45000	1369	642	596	475	356	1.2547
50000	1456	677	628	504	375	1.2460

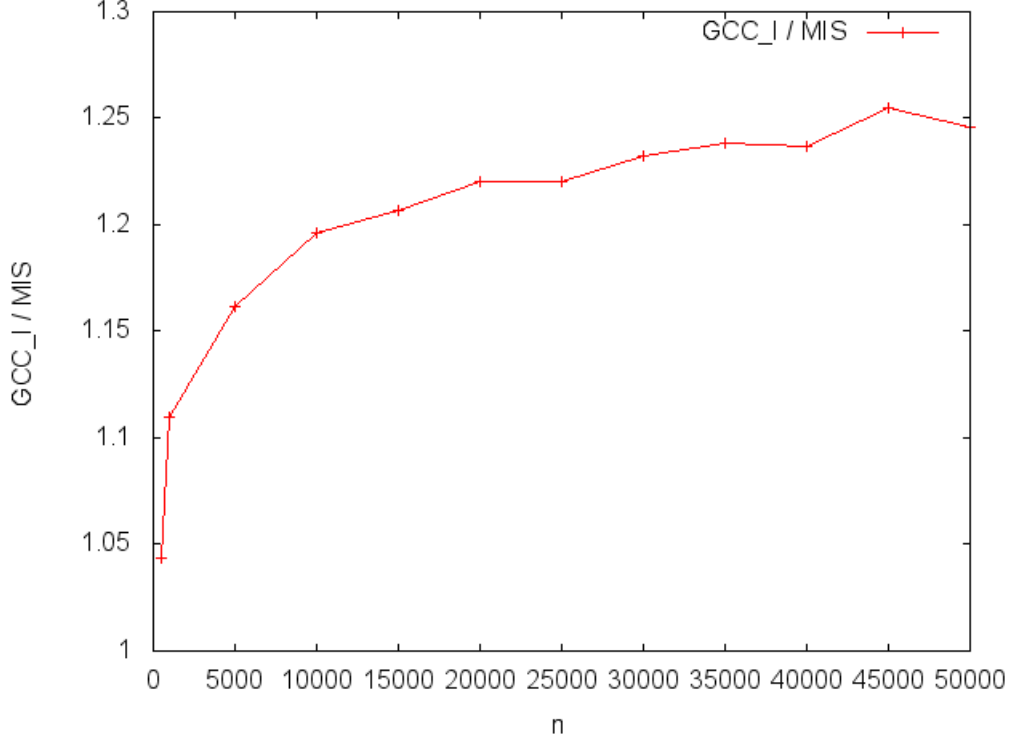


Figure 5: Justification of Observation 1

the chosen values of n , where $|GCC_I|$ and $|MIS|$ are the size of the solution generated by our greedy heuristics GCC_I and MIS for the MCC and MIS problems respectively. Again $|OPT_{MCC}| \leq |CC|$ for any arbitrary clique cover CC and $|OPT_{MIS}| \geq |IS|$ for any arbitrary independent set IS of the given graph. Thus we have

$$OPT_{MIS} \leq OPT_{MCC} < |CC| < 2 \times |IS| \leq 2 \times OPT_{MIS}.$$

Justification 2: In our experiment it is observed that $\frac{|MIS_I|}{|GCC_I|} < 1$ for all the values of n we have chosen. During the execution of GCC_I we observed two parameters Φ and Ψ , where Φ indicates the number of simplicial rectangles observed, and Ψ indicates that the number of times we need to eliminate the largest clique without getting a simplicial rectangle. It is sure that the simplicial rectangles need to be stabbed by a point; so Φ many points are essential. Ψ is the set of extra points used to stab the rectangles that are not stabbed by any point in Φ , and Ψ is less than Φ . So, this gives an indication towards 2 factor approximation algorithm for the MIS problem on the randomly generated rectangle intersection graph. The indication could be well justified if after the elimination of each clique corresponding to a member in Ψ , we could get a simplicial rectangle. But the execution trace (which is not included in this note) does not demonstrate this fact.

Justification 3: It is noticed that since the solution produced by both GCC and GCC_I algorithms for the clique cover problem is very close to $2\sqrt{n}$. It is proved in [13] that the size of the optimum solution of the MCC problem is upper bounded by \sqrt{n} for a set of n randomly positioned rectangles. Thus, the empirical evidences show that our algorithm produces 2 approximation result for large values of n .

5 Conclusion

In this note, we experimentally analyze the performance of greedy algorithm for the minimum clique cover and maximum independent sets problems for rectangle intersection graphs. Experimental result shows that it produces 1.5 factor approximation on the randomly generated instances of the corresponding problems. The intuitive justifications of such behavior may lead to a formal algorithm of getting a constant factor approximation results of the corresponding problems for random instances of rectangle intersection graph.

References

- [1] P. K. Agarwal, M. J. van Kreveld, and S. Suri. Label placement by maximum independent set in rectangles. *Comput. Geom.*, 11(3-4):209–218, 1998.
- [2] B. Aronov, E. Ezra, and M. Sharir. Small-size epsilon-nets for axis-parallel rectangles and boxes. In *STOC*, pages 639–648, 2009.
- [3] P. Chalermsook and J. Chuzhoy. Maximum independent set of rectangles. In *SODA*, pages 892–901, 2009.
- [4] M. Chlebik and J. Chlebikov. Approximation hardness of optimization problems in intersection graphs of d -dimensional boxes. In *SODA*, pages 267–276, 2005.
- [5] V. Chvátal. A greedy heuristic for the set-covering problem. *SIAM Journal on Computing*, 12:759–776, 1983.
- [6] R. J. Fowler, M. Paterson, and S. L. Tanimoto. Optimal packing and covering in the plane are np-complete. *Inf. Process. Lett.*, 12(3):133–137, 1981.
- [7] M. C. Golumbic. *Algorithmic Graph Theory and Perfect Graphs (Annals of Discrete Mathematics, Vol 57)*. North-Holland Publishing Co., Amsterdam, The Netherlands, The Netherlands, 2004.
- [8] M. M. Halldórsson and J. Radhakrishnan. Greed is good: approximating independent sets in sparse and bounded-degree graphs. In *STOC*, pages 439–448, 1994.
- [9] H. Imai and T. Asano. Finding the connected components and a maximum clique of an intersection graph of rectangles in the plane. *J. Algorithms*, 4(4):310–323, 1983.
- [10] J. Mestre. Greedy in approximation algorithms. In *ESA*, pages 528–539, 2006.
- [11] S. C. Nandy and B. B. Bhattacharya. A unified algorithm for finding maximum and minimum object enclosing rectangles and cuboids. *Computers and Mathematics with Applications*, 29(8):45–61, 1995.
- [12] G. L. Nemhauser and W. T. Trotter. Vertex packing: structural properties and algorithms. *Mathematical Programming*, 8:232–248, 1975.
- [13] F. Nielsen. Fast stabbing of boxes in high dimensions. *Theor. Comput. Sci.*, 246(1-2):53–72, 2000.
- [14] F. Nielsen. On point covers of c -oriented polygons. *Theor. Comput. Sci.*, 263(1-2):17–29, 2001.
- [15] J. Pach and G. Tardos. Tight lower bounds for the size of epsilon-nets. In *Symposium on Computational Geometry*, pages 458–463, 2011.
- [16] E. R. Scheinerman. Random interval graphs. *Combinatorica*, 8:357–371, 1980.
- [17] L. V. Tran. Piercing random boxes. *Random Structures & Algorithms*, 38:365–380, 2011.